Rule CIC109: CICS region is approaching maximum capacity

Finding: CPExpert has detected that the amount of CPU used by the quasi-reentrant (QR) mode TCB is approaching the CPU capacity of the region.

Impact: This finding has no direct impact on the performance of the CICS REGION. However, the finding provides an "early warning" that the region is approaching maximum capacity.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: CICS always has two or three TCBs for normal processing, depending on the release of CICS:

- The quasi-reentrant (**QR** mode) TCB executes the quasi-reentrant application code and most CICS code. This TCB is available in all releases of CICS.
- The resource-owning (**RO** mode) TCB is used for opening and closing data sets and for program loading. This TCB is available in all releases of CICS.
- The file-owning (**FO** mode) TCB is used for opening and closing data sets. This TCB is available beginning with CICS/Transaction Server for OS/390, Release 1.

CICS optionally has other TCBs, depending on the release of CICS.

If the Monitoring Class option is ON (MNPER="ON" in the System Initialization Table), the dispatcher domain maintains an "accumulated CPU TCB time" in the DSGACT variable.

Additionally, the dispatcher maintains "accumulated time dispatched" and "accumulated time in MVS wait" variables (DSGTDT and DSGTWT, respectively). The sum of the *accumulated time dispatched* and *accumulated time in MVS wait* is approximately the elapsed time that CICS was operational.

Dividing the *accumulated CPU TCB time* by the sum of the *accumulated time dispatched* and *accumulated time in MVS wait* yields an approximation of the **percent CPU busy** of the CICS region.

The DSGACT value does not include uncaptured CPU time, so the value normally will be less than the amount of CPU time actually used by the CICS region. IBM's *CICS Performance Guides* state that even with a totally busy CICS region, the calculated percent CPU busy of the CICS region would not normally be 100%. Consequently, IBM suggests that a region should be considered approaching maximum capacity if the calculated percent CPU busy exceeds 70%.

CPExpert computes the percent CPU Busy by the following algorithm:

$$\textit{Percent CPU Busy for CICS region} = \frac{DSGACT}{DSGTDT + DSGTWT}$$

where:

DSGACT = Accumulated CPU TCB time
DSGTDT = Accumulated time dispatched
DSGTWT = Accumulated time in MVS wait

CPExpert produces Rule CIC109 when the Percent CPU Busy is greater than the value specified by the **PCTQRTCB** guidance variable in USOURCE(CICGUIDE). The default value for the **PCTQRTCB** guidance variable is 60% indicating that CPExpert should produce Rule CIC109 whenever more than the Percent CPU Busy was more than 60% for the CICS region. The default guidance value is less than the 70% suggested by IBM to give an "early warning" of a capacity restraint.

Suggestion: CPExpert suggests that you review the information provided by Rule CIC109 and determine whether the Percent CPU Busy is reasonable, and is stable. If the Percent CPU Busy is increasing over time, you should consider the following alternatives:

- Review the applications to determine whether the volume of data is increasing and whether the increase is reasonable.
- Review the applications to determine whether coding errors (such as looping) or inefficiencies in design cause increased CPU usage.
- If you have more than one CPU, consider splitting the CICS region and allocating the workload between the resulting CICS regions.
- You can modify the **PCTQRTCB** guidance variable to control the percent CPU busy at which Rule CIC109 is produced.

Reference: *CICS/ESA Version 4.1.1 Performance* Guide: Section 2.2.7.1 and Appendix A.1.4.

CICS/TS Release 1.1 Performance Guide: Section 2.2.7.1 and Appendix 1.1.3.

CICS/TS Release 1.2 Performance Guide: Section 2.2.8.1 and Appendix 1.1.4.

CICS/TS Release 1.3 Performance Guide: Section 2.2.8.1 and Appendix 1.1.5.

CICS/TS for z/OS Release 2.2 *Performance Guide*: Section 2.2.7 and Appendix 1.5.

Thanks: Thanks to Rexaldo Avendano (Kaiser Permanente) for suggesting this rule.

